

**①9 BUNDESREPUBLIK  
DEUTSCHLAND**



**DEUTSCHES  
PATENT- UND  
MARKENAMT**

**Patentschrift**  
**DE 44 34 895 C 2**

(21)	Aktenzeichen:	P 44 34 895.9-53
(22)	Anmeldetag:	29. 9. 94
(43)	Offenlegungstag:	29. 6. 95
(45)	Veröffentlichungstag der Patenterteilung:	24. 12. 98

Int. Cl.<sup>6</sup>:  
**G 06 F 9/30**  
G 06 F 7/52

**DE 44 34 895 C 2**

**Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden.**

**(30) Unionspriorität:**  
173836                      23. 12. 93    US

**(73) Patentinhaber:**  
Hewlett-Packard Co., Palo Alto, Calif., US

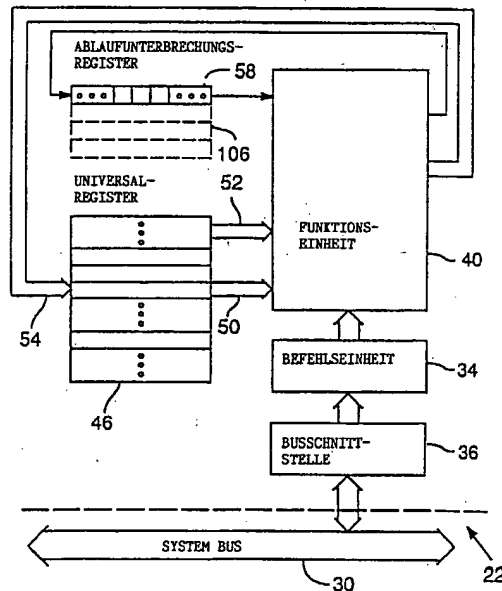
**(74) Vertreter:**  
Schoppe, F., Dipl.-Ing.Univ., Pat.-Anw., 81479  
München

**(72) Erfinder:**  
Worley, William S. jun., Saratoga, Calif., US; Huck,  
Jerome C., Palo Alto, Calif., US; Gupta, Rajiv, Los  
Altos, Calif., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht  
 gezogene Druckschriften:  
 US 51 93 158  
 US 48 75 160

⑤4 Verfahren und Vorrichtung zur Behandlung von Ausnahmebedingungen

Computer, der folgende Merkmale aufweist:  
einen Speicher (24);  
ein Ergebnisregister (88) und ein Ausnahmebedingungsregister (58); und  
eine zentrale Verarbeitungseinheit (22), mit einem vordefinierten Satz von Befehlen, der einen potentiell eine Ausnahmebedingung verursachenden Befehl (68) und einen bedingten Substitutionsbefehl (92) umfaßt, wobei der bedingte Substitutionsbefehl (92) einen Substitutionswert bestimmt, der in einem Ergebnisregister (88) abzulegen ist, wenn in dem Ausnahmebedingungsregister (58) ein Ausnahmebedingungsbit gesetzt ist;  
wobei die zentrale Verarbeitungseinheit (22) auf folgende Ereignisse reagiert:  
(a) auf logische und arithmetische Befehle, die in dem Speicher (24) gespeichert wurden, durch Wiedergewinnen von Daten aus dem Speicher (24), durch Verarbeiten der Daten, durch Erzeugen eines Ergebnisses, und durch Speichern des Ergebnisses in dem Ergebnisregister (88),  
(b) auf das Auftreten einer Ausnahmebedingung durch den potentiell eine Ausnahmebedingung verursachenden Befehl (68) durch Setzen des Ausnahmebedingungsbits in dem Ausnahmebedingungsregister und durch Ausführen des bedingten Substitutionsbefehls (92), und  
(c) auf den bedingten Substitutionsbefehl (92) durch Speichern des Substitutionswertes in dem Ergebnisregister (88) anstelle eines Wertes, der sich in dem Ergebnisregister (88) befindet, wenn in dem Ausnahmebedingungsregister (58) zu dem Zeitpunkt, zu dem der Substitutionsbefehl ausgeführt wird, das Ausnahmebedingungsbit gesetzt ist.



## Beschreibung

Diese Erfindung bezieht sich allgemein auf die Behandlung von Ausnahmebedingungen in einem Computersystem. Insbesondere bezieht sich diese Erfindung auf ein Verfahren und eine Vorrichtung zur Behandlung von Ausnahmebedingungen, die zur Laufzeit auftreten, deren potentiell-  
 5 Auftreten jedoch vorausgesehen werden kann.

Die meisten Befehle, die von einem Computer ausgeführt werden, führen eine bestimmte Art einer arithmetischen oder logischen Operation auf einem oder mehreren Datenoperanden durch und erzeugen resultierende Daten. Typischerweise spezifiziert der Befehl ein oder mehrere Register, in denen die Datenoperanden gespeichert werden, und ein Register, in dem das Ergebnis der Operation gespeichert werden soll. Einige dieser Befehle können ferner Nebenwirkungen oder Ablaufunterbrechungen als eine Folge ihrer Operationen erzeugen, die gewöhnlich nicht im gespeicherten Ergebnis widerspiegelt werden. Das Auftreten derartiger Ablaufunterbrechungen wird typischerweise durch das Speichern eines Zustandsworts, das eine Mehrzahl von Bits oder Flags enthält, in einem Zustandsregister angezeigt.

Ein Beispiel von Befehlen, die Ablaufunterbrechungen erzeugen können, sind Gleitkommaoperationsbefehle. Typische Ablaufunterbrechungen, die eine Folge eines Gleitkommaoperationsbefehls sein können, sind Teilen-durch-Null-, ungenaue, ungültige, Überlauf- und Unterlauf-Ablaufunterbrechungen. Überlauf- und Unterlauf-Ablaufunterbrechungen werden z. B. erzeugt, wenn ein Gleitkomma-Additions-, -Subtraktions-, -Multiplikations- oder -Divisions-Befehl eine Zahl erzeugt, die jenseits des begrenzten Bereichs der darstellbaren Gleitkommazahlen liegt. Teilen-durch-Null-, ungenaue und ungültige Ablaufunterbrechungen werden erzeugt, wenn ein Gleitkommaoperationsbefehl, wie z. B. ein Division-Durch-Null-Operand, eine undefinierte Zahl zur Folge hat.

Üblicherweise ist es nicht möglich, eine Programmausführung fortzusetzen, ohne eine korrigierende Handlung durchzuführen, wenn eine Ablaufunterbrechung als ein Ergebnis eines Befehls erzeugt wird. Z. B. ist das Ergebnis des Befehls eine undefinierte Zahl, wenn ein Gleitkomma-Divisionsbefehl eine Teilen-Durch-Null-Ablaufunterbrechung erzeugt. Nachfolgend ausgeführte Befehle, die das Gleitkomma-Divisionsbefehls-Ergebnis verwenden, würden falsche oder sinnlose Daten erzeugen.

Daher wurde eine Vielfalt von Vorrichtungen verwendet, um auf das Auftreten von Ablaufunterbrechungen anzusprechen. Die einfachste, jedoch durchgreifendste Antwort ist eine Hardwarevorrichtung, die eine Programmausführung einfach beendet oder abbricht, wenn eine Ablaufunterbrechung auftritt. Diese Art einer Hardwarevorrichtung verhindert, daß das Ergebnis des unterbrechenden Befehls verwendet wird, um falsche Daten zu erzeugen, da die Programmausführung nicht fortgesetzt wird. Jedoch sind die Daten, die von dem Programm bis zu dem Auftreten der Ablaufunterbrechung erzeugt werden, typischerweise verloren. Wenn die Programmausführung abgebrochen wird, kann es ferner schwierig sein, zu identifizieren, welcher Befehl und welche Ablaufunterbrechung bewirkt haben, daß die Programmausführung abgebrochen wird.

In den meisten Fällen können Ablaufunterbrechungen allerdings gehandhabt werden, indem eine bestimmte korrigierende Aktion stattfindet, so daß die Programmausführung ohne Verlust von Daten fortgesetzt oder angehalten werden kann. Typische korrigierende Aktionen umfassen das Anzeigen einer Fehlermeldung, es dem Benutzer zu ermöglichen, andere Eingangsdaten einzugeben, bevor eine Berechnung wiederholt wird, oder das teilweise Anhalten der Pro-

grammausführung, während die Daten gesichert werden und identifiziert wird, warum das Programm angehalten wurde.

In einigen Fällen ist es ferner möglich, die Programmausführung nach einem Ablaufunterbrechungsbefehl fortzusetzen, während die Erzeugung falscher Ergebnisse bei nachfolgenden Befehlen vermieden wird, indem ein vorgegebenes Ergebnis für das Ergebnis des Ablaufunterbrechungsbefehls substituiert wird. Z. B. ist bei einigen Berechnungen, die eine Gleitkomma-Divisionsoperation einschließen, für den Division-Durch-Null-Operanden ein Ergebnis definiert. Als ein spezifisches Beispiel schließt die Funktion  $\sin(x)/x$  die Division des Werts von  $\sin(x)$  durch den Wert von  $x$  mit einem Gleitkomma-Divisionsbefehl ein, wenn sie in einem Computer berechnet wird. Wenn  $x$  Null ist, erzeugt der Gleitkomma-Divisionsbefehl eine Division-Durch-Null-Ablaufunterbrechung, wobei eine undefinierte Gleitkommazahl die Folge ist. Jedoch ist bekannt, daß diese Funktion für  $x$  gleich Null gleich Eins ist. Durch Substituieren von Eins für das undefinierte Ergebnis des Gleitkomma-Divisionsbefehls kann das Programm fortgesetzt werden, und nachfolgende Befehle, die das substituierte Ergebnis verwenden, erzeugen genaue Daten. Vom Standpunkt der Mathematik her ist diese Art von Ablaufunterbrechung als eine beherrschbare Singularität bekannt. Die korrigierende Aktion des Substituierens eines vorgegebenen Ergebnisses kann für weitere Ablaufunterbrechungen verwendet werden, die durch Gleitkommaoperations- und Nicht-Gleitkommaoperations-Befehle erzeugt werden.

Korrigierende Aktionen werden im allgemeinen in Software-routinen, die als Ablaufunterbrechungstreiber bekannt sind, durchgeführt. Die Ablaufunterbrechungstreiber bestehen aus Befehlen, die eine Fehlermeldung anzeigen, teilweise das Programm verlassen, ein vorgegebenes Ergebnis substituieren oder andere korrigierende Aktionen durchführen. Im allgemeinen sind Ablaufunterbrechungstreiber für eine bestimmte Ablaufunterbrechung und für einen bestimmten Befehl in einem Programm spezifisch.

Es gibt zwei übliche Vorrichtungen zum Aufrufen eines Ablaufunterbrechungstreibers, um als Reaktion auf Ablaufunterbrechungen eine korrigierende Aktion durchzuführen. Die Vorrichtungen können mit "Tests und Verzweigungen" ("tests and branches") und "Fallen" ("traps") bezeichnet werden. Tests und Verzweigungen sind eine reine Softwarevorrichtung, die das Einfügen von Befehlen vor einem möglichen Ablaufunterbrechungsbefehl einschließt, um sowohl die Werte der Befehlsoperanden zu testen, als auch zu einem geeigneten Ablaufunterbrechungstreiber zu verzweigen, wenn die Werte der Operanden eine Ablaufunterbrechung bewirken würden. Z. B. können Test- und Verzweigungs-Befehle vor einen Gleitkomma-Divisionsbefehl eingefügt sein, um den Divisor-Operanden zu testen, und um zu einem Ablaufunterbrechungstreiber zu verzweigen, wenn der Divisor-Operand Null ist. Der Ablaufunterbrechungstreiber kann dann eine geeignete korrigierende Aktion durchführen, z. B. ein vorgegebenes Ergebnis substituieren, wenn ein solches Ergebnis definiert ist, und nach dem Gleitkomma-Divisionsbefehl die Programmausführung wieder aufnehmen. Der Lösungsansatz des Platzierens von Test- und Verzweigungs-Befehlen ist speziell in Computern nützlich, die keine Hardwarevorrichtung zum Behandeln von Ablaufunterbrechungen liefern, oder die einfach die Programmausführung beenden, wenn eine Ablaufunterbrechung erzeugt wird.

Eine Falle ist eine Hardwarevorrichtung, die in mancher Hinsicht extern erzeugten Unterbrechungen ähnlich ist. In Computern mit einer Fallenvorrichtung ist ein Satz von Speicherplätzen für den Zweck des Speicherns der Startadressen der Ablaufunterbrechungstreiber reserviert oder bestimmt. Diese gespeicherten Startadressen werden mit

Fallenvektoren bezeichnet. Im allgemeinen gibt es einen Fallenvektor für jeden Typ einer Ablaufunterbrechung, die durch den Computer erzeugt werden kann. Wenn eine Ablaufunterbrechung auftritt, hält die Fallenvorrichtung die Programmausführung temporär an und überträgt die Ausführung von dem Programm zwangsläufig zu dem Ablaufunterbrechungstreiber an dem entsprechenden Fallenvektor für die Ablaufunterbrechung. Der Computer führt dann den Ablaufunterbrechungstreiber aus, um eine korrigierende Aktion durchzuführen. Informationen, die die Programmausführung (den Programmkontext) betreffen, wie z. B. die Adressen, an denen die Programmausführung unterbrochen wurde, und die gegenwärtigen Inhalte der Computerregister, werden typischerweise in einem Stapelspeicher gespeichert, wenn die Falle erzeugt wird. Die Speicherung des Programmkontexts ermöglicht eine Wiederaufnahme der Programmausführung, nachdem die Ausführung des Ablaufunterbrechungstreibers abgeschlossen ist. Dieser Typ von Programmunterbrechung, der eine Wiederaufnahme des Programms an einem bestimmten Punkt ermöglicht, ist als eine präzise Unterbrechung bekannt.

Sowohl die Test-und-Verzweigungs- und die Fallen-Vorrichtung haben den Nachteil, daß sie zur Komplexität der Software hinzukommen und die Programmausführung verlangsamen. Bei der Test-und-Verzweigungs-Vorrichtung muß der Programmierer mehrere Befehle vor jedem möglichen Ablaufunterbrechungsbefehl einfügen, um auf verschiedene Daten zu testen, die eine Ablaufunterbrechung erzeugen könnten, und um auf geeignete Ablaufunterbrechungstreiber zu verzweigen. Das Einfügen der Test-und-Verzweigungs-Befehle kompliziert die Programmieraufgabe und addiert sich zur Größe des Programms. Wenn sich der mögliche Ablaufunterbrechungsbefehl in einer Schleife oder einer anderen wiederholten Routine befindet, bringen die hinzugefügten Test- und Verzweigungs-Befehle bei jeder Iteration der Schleife einen Geschwindigkeitsnachteil mit sich, der die Geschwindigkeit der Programmausführung stark beeinflusst.

Bei der Fallenvorrichtung wird ein Geschwindigkeitsnachteil nur aufgebürdet, wenn durch den Ablaufunterbrechungsbefehl tatsächlich eine Ablaufunterbrechung erzeugt wird. Jedoch kann der Geschwindigkeitsnachteil noch signifikant sein, da präzise Unterbrechungen eine Zeitverzögerung bewirken. Das Programmieren der Ablaufunterbrechungstreiber kann ferner komplexer sein, insbesondere wenn die korrigierende Aktion darin besteht, einfach ein vorgegebenes Ergebnis zu substituieren. Da das Ablaufunterbrechungsbefehls-Ergebnis üblicherweise in einem Register gespeichert wird und die Registerinhalte zusammen mit dem Programmkontext in dem Stapelspeicher gespeichert werden, kann die Substitution des vorgegebenen Ergebnisses das Speichern des Substitutionsergebnisses an dem richtigen Platz des Stapelspeichers erfordern. Dies kann eine Anzahl von "Push and Pop"-Operationen erfordern, um den Stapelspeicher richtig zu manipulieren.

Diese beiden Vorrichtungen bringen ferner Probleme bei Pipeline- und Vektor-Computern mit sich, die durch Überlappen oder gleichzeitige Befehlsausführung eine schnellere Programmausführung erreichen. In einem Pipeline-Computer werden Befehle in einer Fließbandform ausgeführt. Jeder Befehl wird in einer Reihe von Stufen ausgeführt. Mehrere Befehle werden an verschiedenen Ausführungsstufen gleichzeitig ausgeführt. Manche Pipeline-Computer verwenden z. B. drei Hauptstufen: Lesen, Ausführen und Schreiben. Jede Stufe kann ferner eine Anzahl von Unterstufen haben. In der Lesestufe werden der Befehl und seine Operanden vom Computer aus dem Speicher oder den Registern gelesen. In der Ausführungsstufe wird für den Befehl

eine Operation auf den Operanden durchgeführt. In der Schreibstufe wird das Ergebnis der Operation formatiert und zur Register- oder Speicher-Speicherung übertragen. Während die Ausführungsstufe eines Befehls durchgeführt wird, ist ein vorheriger Befehl im allgemeinen in seiner Schreibstufe und ein nächster Befehl ist in seiner Lesestufe. Da in einem Pipeline-Computer mehr als ein Befehl gleichzeitig ausgeführt werden kann, läuft ein Programm im allgemeinen viel schneller als in einem Nicht-Pipeline-Computer.

In einem Pipeline-Computer können nicht alle Befehle eines Programms gleichzeitig ausgeführt werden. Das liegt gewöhnlich daran, daß es eine gewisse Datenabhängigkeit zwischen den Befehlen gibt. Z. B. wird eine Situation, die als Lesen-Nach-Schreiben-Gefahr bekannt ist, erzeugt, wenn das Ergebnis eines ersten Befehls als ein Operand eines nachfolgenden Befehls verwendet wird. Da das Ergebnis nicht verfügbar ist, um gelesen zu werden, bis es geschrieben ist, kann der nachfolgende Befehl nicht gleichzeitig mit dem ersten Befehl ausgeführt werden. Die Lesestufe des nachfolgenden Befehls muß der Schreibstufe des ersten Befehls folgen. Datenabhängigkeiten können durch Verschieben der Reihenfolge der Befehle, wenn es möglich ist, oder durch Einfügen von Null-Befehlen zwischen datenabhängigen Befehlen während des Kompilierens eines Programms vermieden werden. Ferner kann Hardware, die zur Laufzeit nach Datenabhängigkeiten überprüft und Null-Befehle zwischen datenabhängigen Befehlen einfügt, verwendet werden.

Test-und-Verzweigungs-Vorrichtungen sind in einem Pipeline-Computer nachteilig, da sie inhärent eine gleichzeitige Befehlsausführung blockieren. Jedesmal, wenn eine gemeinsame Ausführung blockiert wird, wird die Programmausführung verlangsamt. Bis ein Verzweigungsbefehl ausgeführt ist, ist nicht bekannt, welcher der zwei Befehle – der nächste Befehl in der Abfolge oder ein Fernbefehl – als nächstes ausgeführt werden soll. Eine gemeinsame Befehlsausführung ist daher nicht möglich. Dieser Nachteil kann durch Hardware teilweise verschoben werden, die versucht, vorherzusagen, welcher der zwei Befehle dem bedingten Verzweigungsbefehl folgen wird, und beginnt, den vorhergesagten Befehl gleichzeitig auszuführen. Jedoch ist diese Vorhersage nicht immer genau. Wenn sich die Vorhersage als ungenau erweist, muß die Hardware die Ausführung des ungenau vorhergesagten Befehls rückgängig machen und beginnen, den anderen Befehl auszuführen. Folglich wird ein weiterer Geschwindigkeitsnachteil erzeugt, wenn eine Verzweigung nicht genau vorhergesagt wird. Eine derartige hinzugefügte Hardwarefunktionalität addiert sich zu den Kosten und der Komplexität des Pipeline-Computers.

Die Fallenvorrichtung besitzt ebenfalls den Nachteil des Unterbrechens der gemeinsamen Ausführung und des Addierens zu der Komplexität und den Kosten der Hardware in einem Pipeline-Computer. Um nach der Übertragung der Ausführung zu einem Ablaufunterbrechungstreiber erfolgreich zu einer Programmausführung zurückzukehren, muß eine Fallenvorrichtung eine "präzise Unterbrechung" durchführen, bei der der Zustand der Programmausführung bis zu dem Ablaufunterbrechungsbefehl gesichert wird, um eine Wiederaufnahme der Programmausführung am Punkt der Unterbrechung zu ermöglichen. Da in einem Pipeline-Computer mehrere Befehle gleichzeitig ausgeführt werden, werden einige der Befehle, die dem Ablaufunterbrechungsbefehl folgen, in verschiedenen Ausführungsstufen sein. Einige werden bereits die Daten in dem Register des Computers oder die Zustandsbits in dem Ablaufunterbrechungsregister geändert haben. Um die Programmausführung bei dem Ablaufunterbrechungsbefehl anzuhalten, muß die Fal-

lenvorrichtung in der Lage sein, die Pipeline weiterer gleichzeitiger Ausführungsbefehle "abzuleiten". Das Ableiten der Pipeline kann das Wiederaufrufen dessen, was andere Befehle nach dem Ablaufunterbrechungsbefehl durchgeführt haben, und das Rückgängigmachen ihrer Wirkungen erfordern. Eine solche Hardware, die präzise ihrer Pipeline-Befehle unterbrechen kann, ist komplexer und aufwendiger, da sie in der Lage sein muß, mehrere gleichzeitige Befehle zu bedenken und rückgängig zu machen.

Einige Pipeline-Computer vereinfachen die Fallenvorrichtung, indem sie erfordern, daß ein Sperroperationsbefehl nach einem möglichen Ablaufunterbrechungsbefehl eingefügt wird. Der Sperroperationsbefehl behindert eine gleichzeitige Ausführung, indem nachfolgende Befehle daran gehindert werden, gleichzeitig mit den möglichen Ablaufunterbrechungsbefehlen ausgeführt zu werden. Dies leitet im wesentlichen die Befehlspipeline ab, um ein Einfangen oder Verzweigen zu einem Ablaufunterbrechungstreiber zu ermöglichen. Der Nachteil der Sperroperationsbefehls-Vorrichtung besteht darin, daß die gleichzeitige Ausführung jedesmal behindert wird, wenn ein Befehl, der eine Ablaufunterbrechung erzeugen könnte, ausgeführt wird.

Die US-A-5,193,158 betrifft ein Verfahren zur Handhabung von Ausnahmebedingungen in Pipeline-Prozessoren, das die Ausführung von Befehlen, die eine Ablaufunterbrechung eines Programms hervorrufen, verhindert. Gleichzeitig werden Befehle, die vor dem Auftreten des den Programmablauf unterbrechenden Befehls noch nicht vollständig abgearbeitet wurden, nicht weiter ausgeführt werden. Die Ausführung wird nach der Beseitigung der Ablaufunterbrechung erneut begonnen.

Die US-A-4,875,160 betrifft ein Verfahren zur Durchführung einer Wiedergewinnung bei Ausnahmebedingungen in einem Pipeline-Prozessor, bei dem "Zustands-Silos" vorgesehen sind, die eine ausreichende Menge an Zustandsinformationen hinsichtlich des Systems beinhalten, um das Computersystem vom Pipeline-Typ nach der Beendigung der Handhabung der Ablaufunterbrechung erneut zu starten.

Es ist die Aufgabe der vorliegenden Erfindung, eine Vorrichtung und ein Verfahren zur Behandlung von Ausnahmebedingungen in einem Computersystem zu schaffen, wobei weder die Geschwindigkeit einer Programmausführung beeinflusst wird, noch übermäßige zusätzliche Kosten und Komplexität eingeführt werden.

Diese Aufgabe wird durch ein Computersystem gemäß Patentanspruch 1 und durch ein Verfahren gemäß Patentanspruch 13 gelöst.

Die vorliegende Erfindung schafft ein Verfahren und eine Vorrichtung zur Erholung von Ablaufunterbrechungen, die die oben genannten und weitere Nachteile bekannter Technik überwinden. Gemäß der Erfindung ist der Befehlssatz eines Computers (der ein Pipeline-Computer sein kann) durch Einschließen eines oder mehrerer "Fix-up"-Befehle, die als Reaktion auf einen Befehl, der eine oder mehrere spezifizierte Ablaufunterbrechungen erzeugt, eine einfache korrigierende Aktion, wie z. B. die Substitution eines vorgegebenen Wertes, bewirken, erweitert.

Bei einem Ausführungsbeispiel der Erfindung ist z. B. ein bedingter Substitutionsbefehl im Befehlssatz eines Pipeline-Computers enthalten. Der bedingte Substitutionsbefehl spezifiziert einen Satz von einer oder mehreren möglichen Ablaufunterbrechungen, einen vorgegebenen Wert und ein Ergebnisregister. Der bedingte Substitutionsbefehl wird nach einem möglichen Ablaufunterbrechungsbefehl, für den die Substitution des vorgegebenen Ergebnisses eine durchgehende Programmausführung ermöglicht, in ein Programm eingefügt. Wenn der mögliche Ablaufunterbrechungsbefehl die spezifizierte Ablaufunterbrechung erzeugt, während er

ausgeführt wird, ersetzt der bedingte Substitutionsbefehl das Ergebnis des Ablaufunterbrechungsbefehls durch den vorgegebenen Wert, der in dem Ergebnisregister gespeichert ist.

Die "Fix-up"-Befehle können mit möglichen Ablaufunterbrechungsbefehlen und anderen Befehlen in einer Pipeline-Form ausgeführt werden. Durch Zustandsbits, die in einem Ablaufunterbrechungsregister als ein Ergebnis des Ausführens des Befehls gespeichert sind, wird angezeigt, ob eine Ablaufunterbrechung durch einen möglichen Ablaufunterbrechungsbefehl erzeugt wurde. Mögliche Ablaufunterbrechungsbefehle speichern die Zustandsbits während ihrer Schreibstufe in dem Ablaufunterbrechungsregister. Der bedingte Substitutionsbefehl wirkt basierend auf den Zustandsbits, die während seiner Schreibstufe in dem Ablaufunterbrechungsregister gespeichert sind, um den spezifizierten vorgegebenen Wert im Ergebnisregister zu speichern, wenn die Zustandsbits anzeigen, daß eine oder mehrere der spezifizierten Ablaufunterbrechungen auftreten. Dies ermöglicht es, daß der bedingte Substitutionsbefehl, der die Ablaufunterbrechung korrigiert, in einem Pipeline-Computer unmittelbar nach dem Ablaufunterbrechungsbefehl folgt, ohne eine Lesen-Nach-Schreiben-Gefahr zu erzeugen.

Bei einem alternativen Ausführungsbeispiel der Erfindung lädt ein bedingter Substitutionsbefehl eine Speichertabelle mit einem vorgegebenen Wert, der für das Ergebnis eines Befehls, der eine oder mehrere spezifizierte Ablaufunterbrechungen erzeugt, substituiert werden soll. Dieser bedingte Substitutionsbefehl wird vor allen möglichen Ablaufunterbrechungsbefehlen, die er korrigieren soll, plaziert. Ergebnisse von Befehlen, die von einer Funktionseinheit durchgeführt werden, werden durch die Speichertabelle geleitet, bevor sie im Speicher oder in den Registern gespeichert werden. Wenn der Befehl eine oder mehrere der spezifizierten Ablaufunterbrechungen erzeugt, wird das Befehlsergebnis zur Speicherung im Speicher oder Register in seinem Pfad durch den gespeicherten vorgegebenen Wert ersetzt. Wenn infolge der Befehlsausführung keine spezifizierte Ablaufunterbrechung erzeugt wurde, fährt das Befehlsergebnis auf seinem Pfad zur Speicherung im Speicher oder Register fort.

Der bedingte Substitutionsbefehl gemäß diesem alternativen Ausführungsbeispiel hat den Vorteil, daß er für eine beliebige Anzahl von nachfolgenden Befehlen, die die Substitution des gleichen vorgegebenen Werts für die gleichen spezifizierten Ablaufunterbrechungen erfordern, nur einmal durchgeführt werden muß. Dieser bedingte Substitutionsbefehl kann z. B. einmal durchgeführt werden, um die Substitution eines möglichen Ablaufunterbrechungsbefehls, der in einer "Schleifen"-Routine wiederholt ausgeführt wird, zu installieren.

Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend unter Bezugnahme auf die beiliegenden Zeichnungen näher erläutert. Es zeigen:

Fig. 1 ein verallgemeinertes Blockdiagramm eines Pipeline-Computersystems gemäß einem bevorzugten Ausführungsbeispiel der Erfindung;

Fig. 2 ein verallgemeinertes Blockdiagramm eines Abschnitts einer Zentralverarbeitungseinheit (CPU; CPU = Central Processing Unit) im Computersystem von Fig. 1;

Fig. 3 ein Datenflußdiagramm einer Funktionseinheit in der CPU von Fig. 2, die einen Gleitkomma-Divisionsbefehl ausführt;

Fig. 4 ein Datenflußdiagramm der Funktionseinheit von Fig. 2, die einen bedingten Substitutionsbefehl ausführt;

Fig. 5 ein Zeitablaufdiagramm, das die gleichzeitige Pipeline-Ausführung der Gleitkomma-Divisions- und der bedingten Substitutions-Befehle in der CPU von Fig. 1 ausführt;

Fig. 6 ein Diagramm eines Maschinencodes für den bedingten Substitutionsbefehl, der in Fig. 4 gezeigt ist;

Fig. 7 ein verallgemeinertes Blockdiagramm eines Abschnitts einer CPU in einem Computersystem gemäß einem weiteren Ausführungsbeispiel der Erfindung; und

Fig. 8 ein verallgemeinertes Blockdiagramm eines Abschnitts einer CPU in einem Computersystem gemäß einem dritten Ausführungsbeispiel der Erfindung.

Gemäß Fig. 1 umfaßt ein Pipeline-Computersystem 20 gemäß einem bevorzugten Ausführungsbeispiel der Erfindung eine Zentralverarbeitungseinheit (CPU) 22, die über einen Mehrfachsignal-Systembus 30 mit einem Speicher 24 und Peripherie-Vorrichtungen 26 verbunden ist. Der Speicher 24 ist vorzugsweise mit dynamischen Direktzugriffsspeicherchips realisiert, kann jedoch auch mit Nur-Lese-Speicherchips oder anderen elektronischen Speichervorrichtungen realisiert sein. Der Speicher 24 speichert Daten und Befehle für ein oder mehrere Programme, die eine gewünschte Aufgabe oder Berechnung auf dem Computersystem 20 realisieren.

Gemäß Fig. 2 ist die CPU 22 wirksam, um die Befehle eines Programms durchzuführen, um die gewünschte Aufgabe oder Berechnung zu vollenden. Die Befehle, die in dem Speicher 24 gespeichert sind, werden in eine Befehlseinheit 34 in der CPU 22 gelesen. Ein Busschnittstellen-Schaltkreis 36 erzeugt die Signale, die notwendig sind, um Daten und Befehle auf dem Systembus 30 von dem Speicher 24 zu lesen und in diesen zu schreiben. Die Befehlseinheit 34 dekodiert die Befehle und erzeugt Steuersignale, die eine Funktionseinheit 40 leiten, um die Befehle auszuführen.

Die Befehlseinheit 34 ist aufgebaut, um auf einen vorbestimmten Satz von Befehlen (der "Befehlssatz") anzusprechen und die Steuersignale zu bilden, die notwendig sind, um die Befehle mit der Funktionseinheit 40 durchzuführen. Unter der Mehrzahl der Befehle in dem Befehlssatz befinden sich logische Operationsbefehle, arithmetische Operationsbefehle und Gleitkommaoperationsbefehle. Der Befehlssatz umfaßt ferner Datenübertragungsbefehle zum Übertragen von Daten zwischen dem Speicher und einem Satz von Universalregistern 46 und Zwischenregistern. (Der Ausdruck Universalregister wird hierin verwendet, um sowohl die Register zu bezeichnen, die Integer- oder Festkomma-Werte speichern, als auch die Register, die Gleitkomma-Werte speichern.) Gemäß der Erfindung umfaßt der Befehlssatz ferner einen Befehl zum Durchführen einer korrigierenden Aktion als Reaktion auf eine Ablaufunterbrechung, vorzugsweise einen bedingten Substitutionsbefehl, der nachfolgend beschrieben wird.

Die Funktionseinheit 40 führt Operationen durch, um die Befehle, die durch die Befehlseinheit 34 geleitet werden, auszuführen. Die Funktionseinheit 40 umfaßt zwei Lesetore 50, 52 zum Lesen von einem oder zwei Operanden, die in einer Operation von Registern, die aus den Universalregistern 46 ausgewählt sind, verwendet. Die Funktionseinheit 40 umfaßt ferner ein Schreibtor 54 zum Schreiben eines Ergebnisses der Operation in ein Register, das aus den Universalregistern 46 ausgewählt ist. Die Funktionseinheit 40 schreibt ferner Ablaufunterbrechungsdaten in ein Ablaufunterbrechungsregister 58, um das Auftreten von Ablaufunterbrechungen während der Ausführung der Operation anzuzeigen. Die Ablaufunterbrechungsdaten können von der Funktionseinheit 40 ferner aus dem Ablaufunterbrechungsregister 58 gelesen werden.

Gemäß Fig. 5 führt die CPU 22 die Befehle aus dem Speicher 24 bei dem bevorzugten Ausführungsbeispiel in einer Pipeline-Form aus. Gemäß Fig. 5 wird jeder Befehl in drei Stufen, einer Lesestufe, einer Ausführungsstufe und einer Schreibstufe ausgeführt, wobei jede Stufe zur Vollendung

im allgemeinen einen Taktzyklus verwendet. Bei den anderen Ausführungsbeispielen der Erfindung können die Befehle in einer beliebigen Anzahl von Stufen ausgeführt werden. Die CPU 22 kann drei unterschiedliche Stufen von drei Befehlen während jedes Taktzyklusses gleichzeitig ausführen. Z. B. kann die Ausführungsstufe eines Befehls gleichzeitig mit der Schreibstufe eines unmittelbar vorhergehenden Befehls und der Lesestufe eines unmittelbar nachfolgenden Befehls ausgeführt werden. Die Pipeline-Befehlsausführung dient dazu, die Geschwindigkeit zu erhöhen, mit der die CPU 22 Befehle ausführen kann.

Gemäß den Fig. 3 und 5 können einige Befehle in dem Befehlssatz, der in der CPU 22 ausführbar ist, Ablaufunterbrechungen erzeugen. Z. B. können Gleitkommaoperationsbefehle Teilen-durch-Null, ungenaue, ungültige, Überlauf- und Unterlauf-Ablaufunterbrechungen erzeugen. Auch andere Befehle als Gleitkommaoperationsbefehle können Ablaufunterbrechungen erzeugen. Z. B. können Datenübertragungsbefehle Ablaufunterbrechungen erzeugen, wie z. B. im Fall einer ungültigen Adresse. Im allgemeinen wird eine Ablaufunterbrechung erzeugt, wenn der Befehl nicht erfolgreich vollendet werden kann, oder ein undefiniertes oder ungültiges Ergebnis erzeugt wird. Beliebige Daten, die eine Folge des Befehls sind, bewirken daher wahrscheinlich Fehler, wenn sie von einem nachfolgenden Befehl verwendet werden.

Z. B. erzeugt ein Gleitkomma-Divisionsbefehl 68 (Fig. 5) eine Division-Durch-Null-Ablaufunterbrechung, wenn sein Divisor-Operand Null ist. Gemäß Fig. 3 wird die Funktionseinheit 40 durch die Befehlseinheit 34 (Fig. 2), die auf den Gleitkomma-Divisionsbefehl 68 anspricht, geleitet, um eine Gleitkomma-Divisionsoperation auszuführen. Während einer Lesestufe 70 (Fig. 5) des Divisionsbefehls 68 liest die Funktionseinheit 40 zwei Gleitkommaoperanden, einen Divident- und einen Divisor-Operanden, aus zwei Registern 74, 75 (Fig. 3) in den Universalregistern 46, die durch den Divisionsbefehl 68 spezifiziert sind. In einer Ausführungsstufe 78 (Fig. 5) des Divisionsbefehls 68 führt die Funktionseinheit 40 eine Gleitkomma-Divisionsoperation mit den Operanden durch und erzeugt ein Quotientenergebnis. Wenn der Divisor-Operand Null ist, ist das Ergebnis der Gleitkomma-Divisionsoperation im allgemeinen undefiniert. Die Funktionseinheit 40 erzeugt daher also eine Division-Durch-Null-Ablaufunterbrechung. Um beliebige Ablaufunterbrechungen, die durch die Ausführung des Divisionsbefehls erzeugt werden, anzuzeigen, schreibt die Funktionseinheit 40 während einer Schreibstufe 84 (Fig. 5) Ablaufunterbrechungsdaten in das Ablaufunterbrechungsregister 58 (Fig. 3). Das Auftreten einer Division-Durch-Null-Ablaufunterbrechung wird vorzugsweise durch das Einstellen eines Zustandsbits an einer vorbestimmten Position 82 des Ablaufunterbrechungsregisters 58 angezeigt. Ferner platziert die Funktionseinheit 40 während der Schreibstufe 84 (Fig. 5) das Ergebnis der Operation in einem ordnungsgemäßen Gleitkomma-Zahlenformat und schreibt das Ergebnis in ein Register 88 (Fig. 3) in den Universalregistern 46, die durch den Divisionsbefehl spezifiziert sind. Wenn eine Ablaufunterbrechung, wie z. B. eine Division-Durch-Null-Ablaufunterbrechung aufgetreten ist, ist das Ergebnis, das von der Funktionseinheit 40 in das Register 88 gespeichert ist, ungültig und erzeugt Fehler, wenn es von nachfolgenden Befehlen verwendet wird.

Gemäß Fig. 5 ist es bei dem Computersystem 20 möglich, durch die Verwendung eines bedingten Substitutionsbefehls 92, der nach einem Ablaufunterbrechungs-Erzeugungsbefehls, dessen Ablaufunterbrechung vorhergesehen und mittels Substitution eines vorgegebenen Ergebnisses korrigiert werden kann, durchgeführt wird, eine Erholung von Ablauf-

unterbrechungen zu schaffen. Wenn z. B. der Gleitkomma-Divisionsbefehl 68 Teil einer Reihe von Befehlen ist, die die Funktion  $\sin(x)/x$  berechnen, kann der bedingte Substitutionsbefehl 92 verwendet werden, um den bekannten Wert der Funktion für  $x$  gleich Null (d. h. eins) zu substituieren, wenn der Divisionsbefehl eine Division-Durch-Null-Ablaufunterbrechung erzeugt. Die Substitution eines bekannten vorgegebenen Ergebnisses durch einen bedingten Substitutionsbefehl kann verwendet werden, um verschiedene andere Ablaufunterbrechungen, wie z. B. Überlauf- und Unterlauf-Ablaufunterbrechungen durch Substitution einer großen negativen oder positiven Zahl und Ablaufunterbrechungen wegen ungültiger Adressen durch Substituieren eines Null-Werts, zu korrigieren.

Gemäß Fig. 6 umfaßt der bedingte Substitutionsbefehl 92 im allgemeinen einen Operationscode (Op-Code) 94, ein Bestimmungsregister-Spezifizierelement 96, ein Quellregister-Spezifizierelement 98 und ein Ablaufunterbrechungs-Spezifizierelement 100. Der Op-Code identifiziert den Befehl für die Befehlseinheit 34 als einen bedingten Substitutionsbefehl. Beim Empfang eines Befehls mit dem Op-Code für einen bedingten Substitutionsbefehl leitet die Befehlseinheit 34 die Funktionseinheit 40, um eine bedingte Substitutionsoperation auszuführen. Das Ablaufunterbrechungs-Spezifizierelement 100 identifiziert eine oder mehrere Ablaufunterbrechungen, die der bedingte Substitutionsbefehl durch Schreiben eines vorgegebenen oder Substitutions-Werts von einem Register, das durch das Quellregister-Spezifizierelement 98 identifiziert ist, in ein Register, das durch das Bestimmungsregister-Spezifizierelement 96 identifiziert ist, korrigiert.

Gemäß den Fig. 4 und 5 liest die Funktionseinheit 40 in einer Lesestufe 104 der Ausführung des bedingten Substitutionsbefehls 92 z. B. ein vorgegebenes Ergebnis aus einem Register 106, das durch das Quellregister-Spezifizierelement 98 identifiziert ist. Um zu erfassen, ob der Divisionsbefehl 68 eine Ablaufunterbrechung oder Ablaufunterbrechungen, die durch das Ablaufunterbrechungs-Spezifizierelement 100 spezifiziert sind, erzeugt hat, werden die Ablaufunterbrechungsdaten im Ablaufunterbrechungsregister 58 gelesen. Da jedoch die Ablaufunterbrechungsdaten nicht vor der Schreibstufe 84 des Divisionsbefehls 68 in das Ablaufunterbrechungsregister 58 geschrieben werden, werden die Ablaufunterbrechungsdaten in der Schreibstufe des bedingten Substitutionsbefehls 92 gelesen. Wenn die Ablaufunterbrechung oder die Ablaufunterbrechungen, die durch das Ablaufunterbrechungs-Spezifizierelement spezifiziert sind, von dem Divisionsbefehl 68 erzeugt wurden, schreibt die Funktionseinheit 40 das vorgegebene Ergebnis in ein Register, das durch das Bestimmungsregister-Spezifizierelement 96 spezifiziert ist. Gewöhnlich spezifiziert das Bestimmungsregister-Spezifizierelement 96 das gleiche Register 88, in das das Ablaufunterbrechungsbefehls-Ergebnis geschrieben wurde, so daß das Ergebnis auf einen definierten vorgegebenen Wert korrigiert wird.

Wiederum gemäß Fig. 5 folgt der bedingte Substitutionsbefehl 92 vorzugsweise unmittelbar nach dem Befehl 68, dessen Ablaufunterbrechung in der Befehlspipeline korrigiert werden soll, so daß eine Wartezeit der Verfügbarkeit des Ergebnisses minimiert ist. Solange jedoch kein Befehl zwischen dem Befehl 68 und dem bedingten Substitutionsbefehl 92 die Ablaufunterbrechungsdaten in dem Ablaufunterbrechungsregister 58 ändert, oder das Ergebnis in das Register 88 schreibt oder liest, kann der bedingte Substitutionsbefehl 92 eine beliebige Anzahl von Befehlen nach dem möglichen Ablaufunterbrechungsbefehl 68 plazierte sein.

Gemäß den Fig. 2 und 6 müssen bei dem Computersystem 20 alle Befehle, die möglicherweise Ablaufunterbre-

chungen erzeugen, in der Lage sein, Ablaufunterbrechungsdaten in ein Register zu speichern, um anzuzeigen, daß eine Ablaufunterbrechung erzeugt wurde. Um zu ermöglichen, daß mögliche Ablaufunterbrechungsbeefehle zwischen dem bedingten Substitutionsbefehl 92 und dem Befehl 68, der sie korrigiert, positioniert sind, können zusätzliche Ablaufunterbrechungsregister 106 (Fig. 2) in der CPU 22 vorgesehen sein. Die Ablaufunterbrechungsdaten der Befehle, die Ablaufunterbrechungen erzeugen, sind in dem Ablaufunterbrechungsregister 58 als eine Vorgabe gespeichert oder müssen eines der zusätzlichen Ablaufunterbrechungsregister 106, in das ihre Ablaufunterbrechungsdaten gespeichert werden, spezifizieren. Wenn zusätzliche Ablaufunterbrechungsregister 106 vorgesehen sind, umfaßt der bedingte Substitutionsbefehl 92 ferner ein Ablaufunterbrechungsregister-Spezifizierelement 108. Der bedingte Substitutionsbefehl 92 und der Befehl 68, dessen Ablaufunterbrechung korrigiert wird, müssen beide das gleiche Ablaufunterbrechungsregister mit den Ablaufunterbrechungsregister-Spezifizierelementen 108 spezifizieren. Ein Befehl, der ein anderes Ablaufunterbrechungsregister verwendet, um eine Ablaufunterbrechung anzuzeigen, kann dann zwischen den bedingten Substitutionsbefehl und den Befehl 68, der korrigiert wird, plazierte werden, ohne die Ablaufunterbrechungsdaten des Befehls 68 zu ändern.

Wiederum gemäß Fig. 5 können einige potentielle Ablaufunterbrechungsbeefehle mehr als einen Unterbrechungstypen erzeugen. Wenn das Korrigieren jedes der verschiedenen Unterbrechungstypen nur die Substitution des gleichen vorgegebenen Ergebnisses erfordert, wird nur ein bedingter Substitutionsbefehl benötigt. Durch das Spezifizieren aller Ablaufunterbrechungen, die in dem Ablaufunterbrechungs-Spezifizierelement 100 des bedingten Substitutionsbefehls mit der gleichen Substitution korrigiert werden können, führt der bedingte Substitutionsbefehl die notwendige Substitution aus, wenn eine beliebige der spezifizierten Ablaufunterbrechungen erzeugt wurde.

Wenn verschiedene Substitutionen erforderlich sind, um die Ablaufunterbrechungen, die von einem möglichen Ablaufunterbrechungsbeefehl erzeugt werden können, zu korrigieren, müssen mehrere bedingte Substitutionsbeefehle nach dem möglichen Ablaufunterbrechungsbeefehl in die Pipeline eingefügt werden. Jeder der bedingten Substitutionsbeefehle korrigiert eine andere Ablaufunterbrechung oder eine Gruppe von Ablaufunterbrechungen, die mit der gleichen Substitution korrigierbar sind. Jeder bedingte Substitutionsbefehl benötigt die gleiche Zeit, um ausgeführt zu werden, wie jeder andere Befehl in dem Pipeline-Computersystem. Daher verlangsamt jeder bedingte Substitutionsbefehl, der zu einem Programm hinzugefügt wird, die Programmausführung. Jedoch ist die Verlangsamung im allgemeinen viel geringer als die, die durch die Test- und Verzweigungs- oder Fallenvorrichtungen in einem Pipeline-Computersystem erzeugt wird, da die gleichzeitige Ausführung nicht beeinflußt ist.

Bezugnehmend auf Fig. 7 umfaßt in einem "Groß"-Computersystem 120 gemäß einem weiteren Ausführungsbeispiel der Erfindung eine CPU 122 eine Mehrzahl von Funktionseinheiten 124, 126 zum gleichzeitigen Ausführen einer Mehrzahl von Befehlen. Die CPU 122 umfaßt eine Befehlseinheit 130, die Befehle von einem Speicher (nicht gezeigt) über einen Systembus 132 und einen Busschnittstellenschaltkreis 134 empfängt. Die Befehlseinheit 130 spricht auf die Befehle eines vorbestimmten Befehlssatzes, der einen bedingten Substitutionsbefehl für eine Ablaufunterbrechungs-Erholung einschließt, an.

Beim Empfangen von Befehlen, gruppiert die Befehlseinheit 130 eine Mehrzahl von Befehlen, die gleichzeitig aus-

geführt werden können, zusammen. Die Befehlseinheit 130 erzeugt dann separate Steuersignale, um jede der Funktionseinheiten 124, 126 zu leiten, um die gruppierten Befehle, einen Befehl in jeder Funktionseinheit, gleichzeitig auszuführen. Jede Funktionseinheit 124, 126 umfaßt Lesetore 138 bis 141 zum Lesen von Operanden aus den Universalregistern 144 und ein Schreiber 148, 149 zum Schreiben des Ergebnisses einer Operation, die mit den Operanden durchgeführt wird, in eines der Register 144. Ein Ablaufunterbrechungsregister wird jedoch nicht benötigt.

In dem Groß-Computersystem 120 wird ein bedingter Substitutionsbefehl gleichzeitig mit dem Befehl, den er korrigiert, ausgeführt. Z. B. kann ein Divisionsbefehl in der Funktionseinheit 124 gleichzeitig mit der Ausführung eines bedingten Substitutionsbefehls in der Funktionseinheit 126 ausgeführt werden. Wenn beim Ausführen des Divisionsbefehls eine Ablaufunterbrechung erzeugt wird, signalisiert die Funktionseinheit 124 der Funktionseinheit 126 das Auftreten der Ablaufunterbrechung auf einem Ablaufunterbrechungs-Signalsbus 152. Wenn keine Ablaufunterbrechung erzeugt ist, schreibt die Funktionseinheit 124 das Ergebnis des Divisionsbefehls in ein spezifiziertes Register 154 der Universalregister 144. Wenn eine Ablaufunterbrechung erzeugt ist, wird der Funktionseinheit 126 zum Schreiben eines Substitutionsergebnisses in das spezifizierte Register 154, das die Ablaufunterbrechung korrigiert, der Vorrang gegeben.

Alternativ können ein Ablaufunterbrechungsregister oder mehrere Ablaufunterbrechungsregister in dem Groß-Computersystem 120 vorgesehen sein, um Ablaufunterbrechungsdaten zur Verwendung durch einen bedingten Substitutionsbefehl in einer nachfolgend ausgeführten Gruppe von Befehlen zu speichern.

Gemäß Fig. 8 umfaßt eine CPU 172 in einem Computersystem 170 gemäß einem dritten Ausführungsbeispiel der Erfindung einen Multiplexer 174 und eine Speichertabelle 176, die in einem Pfad 178 von Befehlsergebnissen von einer Funktionseinheit 180 zu Universalregistern 182 zwischengeschaltet ist. Die CPU 172 umfaßt ferner eine Befehlseinheit 186, die über einen Systembus 188 und einen Busschnittstellenschaltkreis 190 Befehle von einem Speicher (nicht gezeigt) wiedererlangt. Die Befehlseinheit 186 spricht auf die Befehle eines vorbestimmten Befehlssatzes an, der einen bedingten Substitutionsbefehl einschließt.

Bei dem Computersystem 170 initialisiert der bedingte Substitutionsbefehl die Substitution eines vorgegebenen Wertes für das Ergebnis eines Ablaufunterbrechungsbefehls im voraus. Dies geschieht durch das Speichern eines vorgegebenen Wertes für einen Satz von einer oder mehreren Ablaufunterbrechungen in der Speichertabelle 176. Vorzugsweise umfaßt die Speichertabelle 176 eine Mehrzahl von Einträgen, einen für jede Ablaufunterbrechung, die durch Substitution eines vorgegebenen Werts korrigierbar ist. Der bedingte Substitutionsbefehl, der vorzugsweise ein Quellregister-Spezifizierelement und ein Ablaufunterbrechungs-Spezifizierelement umfaßt, weist die CPU 172 an, einen vorgegebenen Wert von einem der Universalregister 182, das durch das Quellregister-Spezifizierelement spezifiziert ist, in die Einträge der Speichertabelle 176, die zu den Ablaufunterbrechungen gehören, die durch das Ablaufunterbrechungs-Spezifizierelement spezifiziert sind, zu speichern. Der bedingte Substitutionsbefehl kann im allgemeinen von der gleichen Form wie der bedingte Substitutionsbefehl 92 (Fig. 6) mit dem Bestimmungsregister-Spezifizierelement sein, wobei das Ablaufunterbrechungsregister-Spezifizierelement wegfällt.

Der Multiplexer 174 wählt zwischen einem Wert, der an einem Ausgangstor 198 als das Ergebnis eines Befehls, der

von der Funktionseinheit 180 ausgeführt wird, vorliegt, und einem vorgegebenen Wert von der Speichertabelle 176. Der ausgewählte Wert hängt davon ab, ob der ausgeführte Befehl eine Ablaufunterbrechung erzeugte, für die durch einen früheren bedingten Substitutionsbefehl ein vorgegebener Wert gespeichert wurde. Wenn keine solche Ablaufunterbrechung aufgetreten ist, wählt der Multiplexer 174 den Wert, der an dem Ausgangstor 198 vorliegt, aus. Wenn eine solche Ablaufunterbrechung aufgetreten ist, wählt der Multiplexer 174 den vorgegebenen Wert für die Ablaufunterbrechung aus der Speichertabelle 176 aus. Der Wert, der durch den Multiplexer 174 ausgewählt wird, wird in dem Universalregister, das durch den ausgeführten Befehl als die Bestimmung für dessen Ergebnis spezifiziert ist, gespeichert.

Bei dem Computersystem 170 kann eine bedingte Substitution für eine Ablaufunterbrechungserholung mit einem einzigen bedingten Substitutionsbefehl installiert werden, der für eine Anzahl von möglichen Ablaufunterbrechungs-befehlen oder einen wiederholt möglichen Ablaufunterbrechungs-befehl anwendbar ist. Z. B. kann in einem Programm, das die Funktion  $\sin(x)/x$  für eine Reihe von Werten  $x = a_1, a_2, \dots, a_n$  (n sei eine Integerzahl größer als 2) berechnet, ein bedingter Substitutionsbefehl einen Eintrag in der Speichertabelle 176 für Division-durch-Null-Ablaufunterbrechungen mit dem vorgegebenen Wert von 1 laden. Ein Divisionsbefehl, der das Verhältnis von  $\sin(x)/x$  in einer Routine berechnet, die für jeden Wert von x wiederholt wird, wird eine Division-durch-Null-Ablaufunterbrechung erzeugen, wenn der Wert von x Null ist. Wenn ein derartiger Divisionsbefehl die Division-durch-Null-Ablaufunterbrechung erzeugt, wählt der Multiplexer 174 den vorgegebenen Wert, 1, aus der Speichertabelle 176, um ihn als Befehlsergebnis zu speichern.

Obwohl die Grundsätze der Erfindung bezugnehmend auf ein bevorzugtes Ausführungsbeispiel beschrieben und dargestellt sind, ist es offensichtlich, daß die Erfindung im Aufbau und im Detail ohne Abweichen von derartigen Grundsätzen modifiziert werden kann. Z. B. ist die Erfindung ferner in Pipeline-Computersystemen anwendbar, die eine Bypass-Vorrichtung einschließen, durch die Daten, die durch einen ersten Befehl geschrieben werden, durch das Register zu seinem Lesetor geleitet werden, um für einen zweiten Befehl mit gleichzeitiger Ausführung verfügbar zu sein. Ferner kann ein bedingter Substitutionsbefehl eine beliebige logische Kombination von Ablaufunterbrechungen, für die ein Substitutionsergebnis in ein Ergebnisregister geschrieben werden soll, spezifizieren.

Um einen Speicher der Ablaufunterbrechungsinformationen für mehrere Befehle zu liefern, ohne daß mehrere Ablaufunterbrechungsregister (Register 106 in Fig. 5) vorliegen, können als ein weiteres Beispiel die Ablaufunterbrechungsinformationen stattdessen in den Universalregistern 108 gespeichert werden, auf die die Ergebnisse der Befehle gerichtet sind. Im Fall der Universalregister für Gleitkommawerte ist das Ergebnis eines Ablaufunterbrechungs-befehls, das in den Registern gespeichert wird, oft ein ungültiger oder ein "Not-A-Number"-Wert (NaN-Wert). Bei typischen Gleitkommawert-Darstellungen stellt eine Anzahl von verschiedenen Bitkombinationen NaN-Werte dar. Durch Zuweisen bestimmter dieser Bitkombinationen für NaN-Werte auf spezifische Ablaufunterbrechungen, kann der Ergebniswert, der durch einen Ablaufunterbrechungs-befehl in einem Universalregister gespeichert ist, verwendet werden, um das Auftreten einer oder mehrerer spezifizierter Ablaufunterbrechungen anzuzeigen.

Im Fall der Universalregister für Integer- oder Festkomma-Werte, sind im allgemeinen Bitkombinationen verwendet, um gültige Werte darzustellen. Somit sind keine

Bitkombinationen verfügbar, um Ablaufunterbrechungsinformationen darzustellen. Für diese Register müssen Speicher für zusätzliche Bits in jedem Register vorgesehen sein, um Ablaufunterbrechungsinformationen darzustellen.

Sind Ablaufunterbrechungsinformationen in den Universalregistern gespeichert, ist es nicht notwendig, daß Befehle, die eine Ablaufunterbrechung zur Folge haben können, ein Ablaufunterbrechungsregister zum Speichern ihrer Ablaufunterbrechungsinformationen spezifizieren. Alle Ablaufunterbrechungsinformationen des Befehls werden als eine der NaN-Bitkombinationen im gleichen Universalregister gespeichert, das er zum Speichern seines Ergebnisses spezifiziert (oder in dem zusätzlichen Bitspeicher, der für diesen Zweck in dem Fall der Festkommaregister vorgesehen ist). Solche möglichen Ablaufunterbrechungsbefehle kann ein bedingter Substitutionsbefehl, der das Universalregister des Ergebnisses (und der Ablaufunterbrechungsinformationen) entweder mit seinem Bestimmungsregister-Spezifizierelement 96 oder dem Ablaufunterbrechungsregister-Spezifizierelement 108 spezifiziert, folgen (und diese korrigieren).

#### Patentansprüche

##### 1. Computer, der folgende Merkmale aufweist:

einen Speicher (24);  
ein Ergebnisregister (88) und ein Ausnahmebedingungsregister (58); und  
eine zentrale Verarbeitungseinheit (22), mit einem vordefinierten Satz von Befehlen, der einen potentiell eine Ausnahmebedingung verursachenden Befehl (68) und einen bedingten Substitutionsbefehl (92) umfaßt, wobei der bedingte Substitutionsbefehl (92) einen Substitutionswert bestimmt, der in einem Ergebnisregister (88) abzulegen ist, wenn in dem Ausnahmebedingungsregister (58) ein Ausnahmebedingungsbit gesetzt ist;

wobei die zentrale Verarbeitungseinheit (22) auf folgende Ereignisse reagiert:

- (a) auf logische und arithmetische Befehle, die in dem Speicher (24) gespeichert wurden, durch Wiedergewinnen von Daten aus dem Speicher (24), durch Verarbeiten der Daten, durch Erzeugen eines Ergebnisses, und durch Speichern des Ergebnisses in dem Ergebnisregister (88),
- (b) auf das Auftreten einer Ausnahmebedingung durch den potentiell eine Ausnahmebedingung verursachenden Befehl (68) durch Setzen des Ausnahmebedingungsbits in dem Ausnahmebedingungsregister und durch Ausführen des bedingten Substitutionsbefehls (92), und
- (c) auf den bedingten Substitutionsbefehl (92) durch Speichern des Substitutionswertes in dem Ergebnisregister (88) anstelle eines Wertes, der sich in dem Ergebnisregister (88) befindet, wenn in dem Ausnahmebedingungsregister (58) zu dem Zeitpunkt, zu dem der Substitutionsbefehl ausgeführt wird, das Ausnahmebedingungsbit gesetzt ist.

##### 2. Computer gemäß Anspruch 1, bei dem der bedingte Substitutionsbefehl (92) ferner folgende Merkmale aufweist:

einen Operationscode (94), der eine vorbestimmte Folge von einem oder mehreren Bits aufweist, um anzuzeigen, daß, außerhalb der Befehle in dem Befehlssatz, der Befehl der bedingte Substitutionsbefehl (92) ist;  
ein Substitutionsergebnis-Spezifizierelement (98) zum Anzeigen des Substitutionsergebnisses; und

ein Ausnahmebedingungs-Spezifizierelement (100) zum Anzeigen der Ausnahmebedingung.

3. Computer gemäß Anspruch 2, bei dem das Substitutionsergebnis-Spezifizierelement (98) ein Substitutionsergebnisregister, das aus einer Mehrzahl von Registern ausgewählt ist, anzeigt, welches das Substitutionsergebnis enthält, wodurch das Substitutionsergebnis indirekt durch das Substitutionsergebnis-Spezifizierelement (98) angezeigt ist.

4. Computer gemäß Anspruch 3, bei dem der bedingte Substitutionsbefehl (92) ferner folgendes Merkmal aufweist:

ein Ergebnisregister-Spezifizierelement (96) zum Anzeigen, welches Register das Ergebnisregister (88) ist.

5. Computer gemäß einem der Ansprüche 1 bis 4, der ferner folgendes Merkmal aufweist:

eine Mehrzahl von Ausnahmebedingungsregistern (58), von denen jedes Ausnahmebedingungsdaten speichert, die anzeigen, ob eine oder mehrere Ausnahmebedingungen das Ergebnis der Ausführung eines Befehls aus dem Befehlssatz durch die zentrale Verarbeitungseinheit (22) sind,

wobei die zentrale Verarbeitungseinheit (22), die auf den bedingten Substitutionsbefehl (92) anspricht, wirksam ist, um das Substitutionsergebnis in dem Ergebnisregister (88), das aus den Registern ausgewählt ist, zu speichern, wenn die Ausnahmebedingungsdaten, die in einem ausgewählten der Ausnahmebedingungsregister (58) gespeichert sind, anzeigen, daß eine Ausnahmebedingung das Ergebnis der Ausführung eines vorherigen Befehls ist.

6. Computer gemäß einem der Ansprüche 1 bis 5, bei dem der bedingte Substitutionsbefehl (92) ferner folgendes Merkmal aufweist:

ein Ausnahmebedingungsregister-Spezifizierelement (108) zum Anzeigen des Ausgewählten der Ausnahmebedingungsregister (58).

7. Computer gemäß einem der Ansprüche 1 bis 6, bei dem die zentrale Verarbeitungseinheit (22) wirksam ist, um den bedingten Substitutionsbefehl (92) in einer Pipeline-Form gleichzeitig mit anderen Befehlen des Befehlssatzes auszuführen.

8. Computer gemäß einem der Ansprüche 1 bis 4, bei dem die zentrale Verarbeitungseinheit (22), die auf den bedingten Substitutionsbefehl (92) anspricht, wirksam ist, um das Substitutionsergebnis in dem Ergebnisregister (88) zu speichern, wenn eine oder mehrere einer Mehrzahl von ausgewählten Ausnahmebedingungen das Ergebnis eines vorher ausgeführten Befehls sind.

9. Computer gemäß Anspruch 1, bei dem die zentrale Verarbeitungseinheit (22) folgende Merkmale aufweist:

einen Satz von Registern (46; 144; 182) zum Speichern von Daten;

eine Befehlseinheit (34; 130; 186) zum Lesen und Dekodieren von Befehlen aus dem Speicher (24);

eine Funktionseinheit (90; 124; 126; 180), die mit den Registern (46; 144; 182) verbunden ist und logische Schaltungen zum Ausführen von Operationen mit Operanden, die aus den Registern (46; 144; 182) gelesen werden, aufweist;

wobei die Befehlseinheit (34; 130; 186) auf den Befehl (68) anspricht und wirksam ist, um die Funktionseinheit (90; 124; 126; 180) zu veranlassen, eine Operation mit einem oder mehreren Operanden, die aus einem oder mehreren Operandenregistern (74, 75), die aus den Registern (46; 144; 182) ausgewählt sind, gelesen werden, auszuführen und das Ergebnis der Operation in



dem Ergebnisregister (88), das aus den Registern (46; 144; 182) ausgewählt ist, zu speichern; und wobei die Befehlseinheit (34; 130; 186) auf den bedingten Substitutionsbefehl (92) anspricht und wirksam ist, um die Funktionseinheit (90, 124, 126; 180) zu veranlassen, ein Substitutionsergebnis in dem Ergebnisregister (88) zu speichern, wenn eine oder mehrere der Ausnahmebedingungen das Ergebnis der Ausführung der Operation sind.

10. Computer gemäß Anspruch 9, bei dem das Ausnahmebedingungsregister (58) Ausnahmebedingungsdaten speichert, die anzeigen, daß eine Ausnahmebedingung durch die Ausführung einer Operation in der Funktionseinheit (40), die auf einen Befehl (68) anspricht, erzeugt wurde; wobei die Befehlseinheit (34), die auf den Befehl (68) anspricht, wirksam ist, um die Funktionseinheit (40) zu veranlassen, eine Operation mit einem oder mehreren Operanden auszuführen, die aus einem oder mehreren Operandenregistern (74, 75), die aus den Registern (46) ausgewählt sind, gelesen werden, ein Ergebnis der logischen Operation in dem Ergebnisregister (88), das aus den Registern (46) ausgewählt ist, zu speichern, und die Ausnahmebedingungsdaten in dem Ausnahmebedingungsregister zu speichern, wobei angezeigt wird, ob eine oder mehrere Ausnahmebedingungen die Folge der Ausführung der Operation sind; und wobei die Befehlseinheit (34), die auf einen bedingten Substitutionsbefehl (92) anspricht, ferner wirksam ist, um die Funktionseinheit (40) zu veranlassen, das Substitutionsergebnis in dem Ergebnisregister (88) zu speichern, wenn durch die Ausnahmebedingungsdaten in dem Ausnahmebedingungsregister angezeigt wird, daß eine oder mehrere Ausnahmebedingungen aufgetreten sind.

11. Computer gemäß Anspruch 9, der ferner folgendes Merkmal aufweist: eine Mehrzahl von Funktionseinheiten (124, 126), wobei jede Funktionseinheit (124, 126) mit den Registern (144) verbunden ist und logische Schaltungen zum Ausführen von Operationen mit Operanden, die aus den Registern (144) gelesen werden, aufweisen; wobei die Instruktionseinheit (130), die auf eine Gruppe von Befehlen, die den bedingten Substitutionsbefehl (92) einschließen, anspricht, wirksam ist, um eine erste der Funktionseinheiten (124) zu veranlassen, eine Operation mit einem oder mehreren Operanden auszuführen, die aus einem oder mehreren Operandenregistern, die aus den Registern (144) ausgewählt sind, gelesen werden, und ein Ergebnis der Operation in dem Ergebnisregister, das aus den Registern (144) ausgewählt ist, zu speichern; und wobei die Befehlseinheit (130), die auf die Gruppe von Befehlen anspricht, ferner wirksam ist, um eine zweite der Funktionseinheiten (126) zu veranlassen, ein Substitutionsergebnis in dem Ergebnisregister (88) zu speichern, wenn eine oder mehrere Ausnahmebedingungen das Ergebnis der Ausführung der Operation in der ersten Funktionseinheit (126) sind.

12. Computer gemäß Anspruch 9, der ferner folgende Merkmale aufweist:

eine Speichertabelle (176) mit einem oder mehreren Einträgen zum Speichern von Substitutionsergebnissen, wobei das Substitutionsergebnis für die Ausnahmebedingungen als Reaktion auf den bedingten Substitutionsbefehl (92) in der Speichertabelle (176) gespeichert ist; und

einen Multiplexer (174), der zwischen die Funktions-

einheit (180) und die Register (182) geschaltet ist, wobei der Multiplexer (174) wirksam ist, um das Ergebnis, das in dem Ergebnisregister gespeichert ist, durch das Substitutionsergebnis zu ersetzen, wenn die Ausführung der Operation eine oder mehrere Ausnahmebedingungen erzeugt hat.

13. Verfahren zur Behandlung von Ausnahmebedingungen in einem Computer (20), mit folgenden Schritten:

Einlesen eines potentiell eine Ausnahmebedingung verursachenden Befehls (68) und eines bedingten Substitutionsbefehls (92), der eine Zielregisteradresse (96) aufweist und einen festgelegten Substitutionswert liefert;

Ausführen des potentiell eine Ausnahmebedingung verursachenden Befehls (68); Setzen eines Ausnahmebedingungsbits (82), wenn der potentiell eine Ausnahmebedingung verursachende Befehl (68) eine Ausnahmebedingung bewirkt, und Ausführen des bedingten Substitutionsbefehls (92); und

bei gesetztem Ausnahmebedingungsbit (82), Schreiben des Substitutionswerts in ein Ergebnisregister (88), das der Zielregisteradresse entspricht.

14. Verfahren gemäß Anspruch 13, bei dem der Schritt des Ausführens des potentiell eine Ausnahmebedingung verursachenden Befehls (68) ferner das Speichern von Ausnahmebedingungsdaten in einem Register umfaßt, wobei angezeigt wird, ob eine oder mehrere Ausnahmebedingungen das Ergebnis der Ausführung des potentiell eine Ausnahmebedingung verursachenden Befehls (68) sind; und

der Schritt des Ausführens des bedingten Substitutionsbefehls (92) die Bestimmung umfaßt, ob zumindest eine der Ausnahmebedingungen ein Ergebnis der Ausführung des potentiell eine Ausnahmebedingung verursachenden Befehls (68) gemäß den Ausnahmebedingungsdaten, die in dem Ausnahmebedingungsregister (58) gespeichert sind, ist.

15. Verfahren gemäß Anspruch 14, bei dem der Schritt des Entscheidens, ob zumindest eine der Ausnahmebedingungen, die aufgetreten ist, nach den Schritten des Durchführens der Operation auftritt.

16. Verfahren gemäß Anspruch 13, das ferner folgenden Schritt aufweist:

Ausführen einer Mehrzahl von bedingten Substitutionsbefehlen (92), wobei jeder der bedingten Substitutionsbefehle (92) ein Substitutionsergebnis erzeugt und eine oder mehrere Ausnahmebedingungen spezifiziert, wobei der Schritt des Ausführens der bedingten Substitutionsbefehle (92) für jeden bedingten Substitutionsbefehl (92) folgende Schritte aufweist:

Bestimmen, ob zumindest eine der einen oder mehreren Ausnahmebedingungen, die durch den bedingten Substitutionsbefehl spezifiziert sind, als ein Ergebnis der Ausführung des ersten Befehls aufgetreten ist; und Ersetzen des Ergebnisses durch das Substitutionsergebnis, das durch den bedingten Substitutionsbefehl (92) erzeugt ist, wenn bestimmt wird, daß zumindest eine der einen oder mehreren spezifizierten Ausnahmebedingungen aufgetreten ist.

17. Verfahren gemäß Anspruch 13, bei dem der Schritt des Ausführens des bedingten Substitutionsbefehls (92) ferner folgenden Schritt aufweist:

Speichern des Substitutionsergebnisses in einer Speichertabelle (176) vor dem Ausführen des potentiell eine Ausnahmebedingung verursachenden Befehls (68);

wobei die Schritte des Bestimmens und Speicherns des

Substitutionswertes in dem Ergebnisregister (88) nach dem Schritt des Durchführens der Operation auftreten.

Hierzu 5 Seite(n) Zeichnungen

5

10

15

20

25

30

35

40

45

50

55

60

65

- Leerseite -

FIG. 1

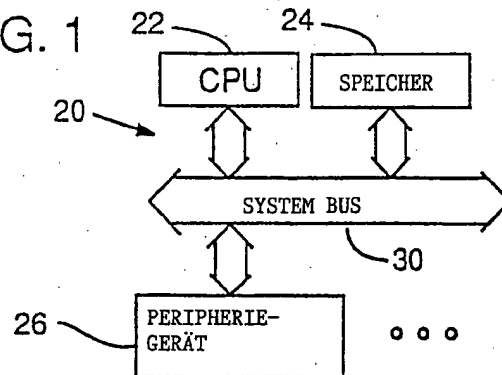
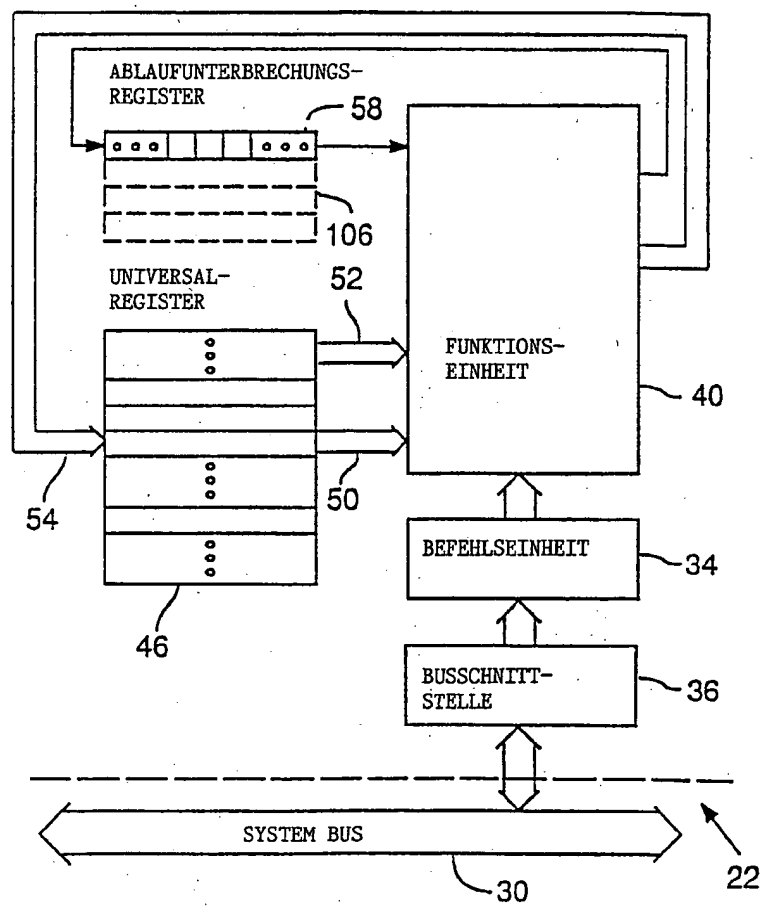


FIG. 2



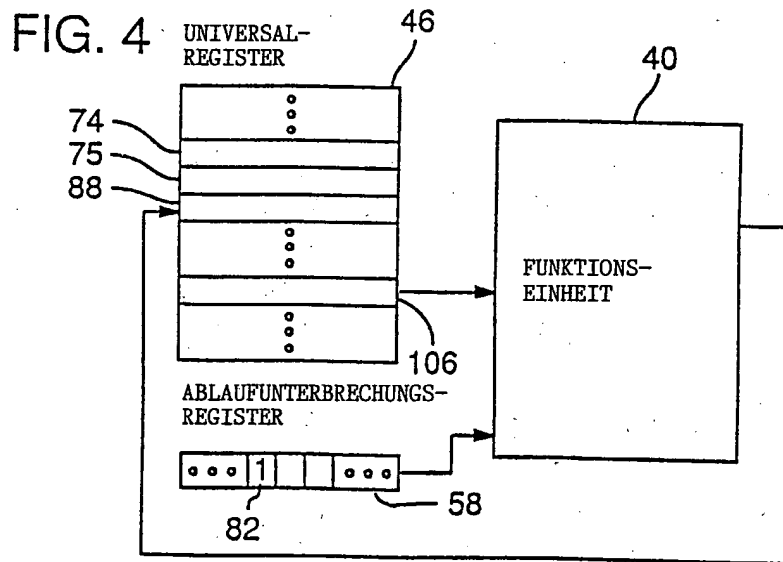
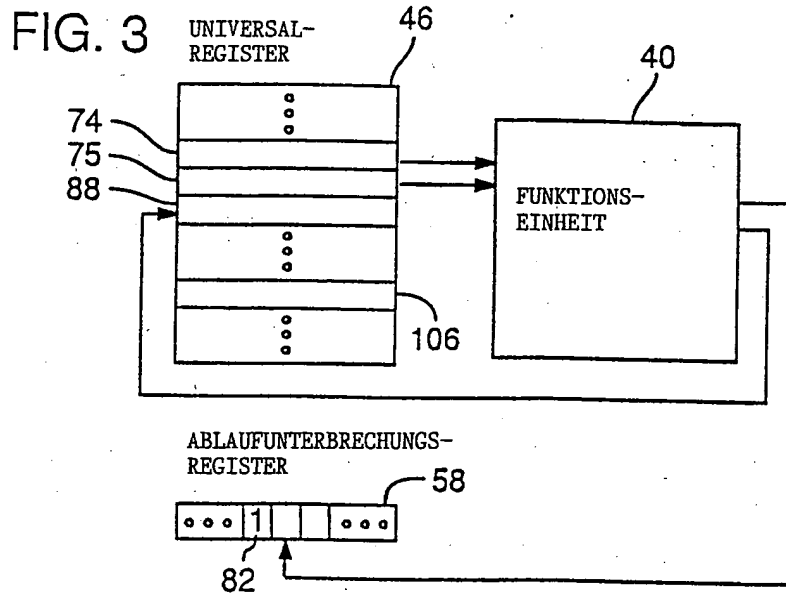




FIG. 7

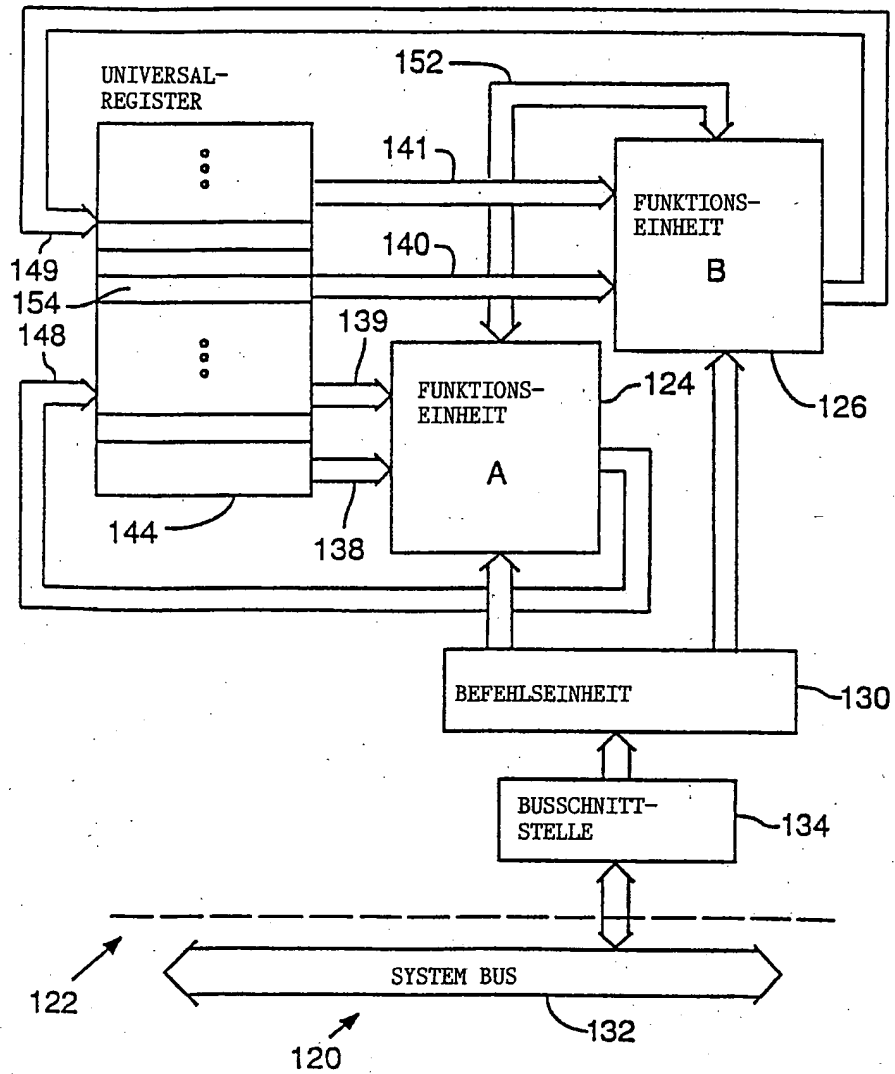


FIG. 8

